Naval Research Laboratory

Washington, DC 20375-5000

NRL Memorandum Report 6780

Tripod Operators for the Interpretation of Range Images

FRANK J. PIPITONE

Navy Center for Applied Research in Artificial Intelligence
Information Technology Division

February 19, 1991

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1991 February 19	

4. TITLE AND SUBTITLE

Tripod Operators for the Interpretation of Range Images

5. FUNDING NUMBERS

RS-34-C74-000
55-0230-0-1
PE 62234N

6. AUTHOR(S)

Frank Pipitone

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL Memorandum Report 6780

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ONT

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

Unlimited

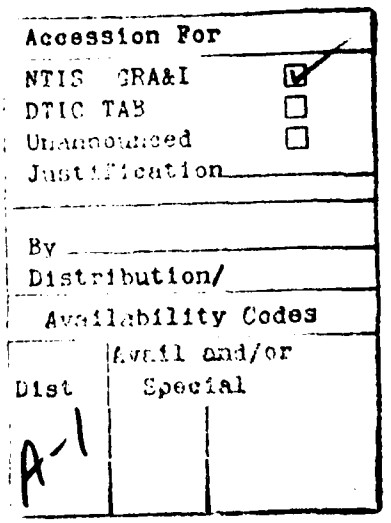
13. ABSTRACT *(Maximum 200 words)*

A new kind of feature extraction operator for range images is introduced that facilitates object recognition in several ways. It consists of three points in 3-space fixed at the vertices of an equilateral triangle and one or more curves, called test curves, fixed in the reference frame of the triangle. This mathematical structure is then moved as a rigid body until the vertices all lie on the surface of some range image or modeled object. The point(s) of intersection of the test curve(s) and the surface are used to define local shape features which are invariant under rigid motions. These features can be used to automatically find distinctive regions at which the begin recognition, to rapidly screen candidate modeled objects for a match, and to speed pruning in the generation of interpretation trees. Tripod operators are applicable to all 3-D shapes, and reduce the need for specialized feature detectors.

14. SUBJECT TERMS		15. NUMBER OF PAGES
Range images	Tripod operators	32
Range data	Interpretation	16. PRICE CODE
Vision	Object Recognition	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

CONTENTS



TRIPOD OPERATORS FOR THE INTERPRETATION OF RANGE IMAGES

1. Introduction

During the past decade, research in the acquisition and use of range images in computer vision has increased greatly. This is due to their relatively complete and explicit representation of 3-D shape information, in contrast to intensity images, from which the recovery of shape is known to be very difficult. Work in this area has led to the development of increasingly fast and accurate rangefinders [6] and to a variety of increasingly effective methods for recognizing and locating modeled objects in range images. The fundamental limits of performance have, however, not nearly been reached. This paper pursues the goal of high speed object recognition by introducing a class of range image operators that extract local shape information that is invariant under rotations and translations of the object with respect to the rangefinder. These operators can be applied to 3-D objects of any shape. They exploit the fact that a small number (e.g., four to six) of range measurements often contain a large amount of information about the identity and pose of objects on which they lie, particularly when the range data is very precise. We will develop the operators in the context of the problem of recognizing and locating modeled rigid 3-D objects in range images, but will suggest other potential applications for them in vision and tactile sensing. The operators arose from studying the problem of efficiently mapping small sets of range measurements into sets of possible object poses. This was achieved by structuring both the range data and the pose representation so that the mapping involves involves sets small enough to compute offline and store. The purpose of this paper is to introduce the tripod operator and describe a wide variety of its properties and potential applications, in the interest of stimulating other work.

The most closely related previous work is by Grimson [1,5], who extensively developed the idea of searching for associations between image features and model elements consistent with geometric constraints among the model elements, using *interpretation trees* to represent the consistent hypothesised associations (interpretations). This general approach was introduced by several authors [2,3,4,10] within a short time. Our work differs from these efforts in that we provide mechanisms for efficiently prestoring model information so that the costly early stages of interpretation tree generation can be avoided at recognition time. In contrast to [5], we use both dense range images and sparse sets automatically chosen from such images.

Manuscript approved December 17, 1990.

1.1 Useful Definitions and Concepts

We will define some terminology and review some concepts in the interest of a concise presentation. Some of the concepts are from previous work on interpretation trees. Suppose we represent a rigid 3-D object by a polyhedron with M facets $\{f_i\}$, $1 \le i \le M$. We denote by $\mathbf{p}_i$ a pixel taken from a range image. We regard a *range pixel* here as simply a point in space measured by a rangefinder, represented by a cartesian three-vector. We define a *range point* as either a range pixel or a point on an interpolated surface between range pixels. A *pairing* is defined as an association of a range pixel with a model facet. This is used to represent the hypothesis that the specified range point actually lies on the region of an object represented by the specified model facet. A set of k pairings will be called a *k-interpretation.* A *k-interpretation* will be called a *partial interpretation* if k is less than the size of some set of range points that we wish to interpret. For example, $\{(\mathbf{p}_1, f_{17}), (\mathbf{p}_2, f_{22}), (\mathbf{p}_3, f_{74})\}$ is a 3-interpretation. Note that if f_{17}, f_{22}, and f_{74} were all infinitely small facets, then the 3-interpretation would imply a precise pose for the object, provided that $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are not colinear, since fixing three noncolinear points belonging to a rigid object prevents the object from moving. A *pose* is defined as a complete specification of the location and orientation of an object, corresponding to an element of the group $\mathbf{R}^3 \otimes SO(3)$. One direct way to do this is by specifying the six coordinates (X,Y,Z,Θ,Φ,Ψ); the three cartesian coordinates of a reference point on the object and three Euler angles, respectively. A second way is to specify the location in space of each of three specified (non-colinear) points on the objects model's surface. There are, of course, many other ways to represent pose.

1.2 Recognition with a Small Number of Range Points; Continuous Analysis

To motivate the introduction of tripod operators, we will consider a problem of recognizing and locating (determining the pose of) a modeled object in a range image using a small number of range points $\mathbf{p}_i$. In this section we will make the argument that a great deal of information about identity and pose is contained in a few points. Later we will exploit this by developing efficient "points to poses" mapping procedures. For the time being, we will assume that the grouping problem is solved, that is, $\mathbf{p}_i$ all lie on the surface of one object. We will also temporarily assume zero uncertainty in both the model and the range measurements. We will later discuss the problems of uncertainty and grouping, since they are crucial in any practical object recognition system. The problem now is to determine what rigid motion(s) of the model, if any, will cause all the $\mathbf{p}_i$ to lie on the surface of the model. Note that since it is the *relative* pose of the model and the range data that is of interest, we will speak sometimes of motions of the model, and sometimes of motions of the range data, according to convenience. We will not yet

invoke a particular representation of 3-D shape. We will now successively impose the constraints that each $\mathbf{p}_i$ lies on the surface of the object and note the effect on our knowledge of the object's pose and identity. Initially the model is free in all six degrees of freedom (DOF). Then, as we successively require each $\mathbf{p}_i$ to lie on the model's surface, successively fewer DOF of motion are available to the model that we are trying to match to the points. That is, the set of possible poses is reduced. If no pose is possible, recognition fails for that model. Usually, introducing each additional point reduces the number of DOF by one. If this is not the case, we say that there are *object symmetries*. We will assume the case of no such symmetries until section 1.3. The following discussion will use the example illustrated in Fig. 1, which depicts six points obtained from a range image of the surface of a solid rectangle for which we have a model.

One Point: If the model is moved into contact with $\mathbf{p}_1$, it has five remaining degrees of freedom; $\mathbf{p}_1$ can lie anywhere on the 2-D surface of the model, and the model is free to rotate about $\mathbf{p}_1$, yielding five DOF.

Two Points: If the model's pose is further constrained by contact with $\mathbf{p}_2$, then if the modeled object is sufficiently large compared to $d_{12} \equiv \| \mathbf{p}_1 - \mathbf{p}_2 \|$, then $\mathbf{p}_1$ can lie anywhere on the 2-D surface of the object. For any such placement, $\mathbf{p}_2$ can then lie anywhere on the space curve formed by the intersection of the model surface and a sphere centered at $\mathbf{p}_1$, constituting a third DOF. For any such placement of two points, the model is free to rotate about the line connecting $\mathbf{p}_1$ and $\mathbf{p}_2$, yielding a total of four DOF. Note that if d_{12} is sufficiently large compared with the model, $\mathbf{p}_1$ and $\mathbf{p}_2$ could not both lie on the model. Hence, two points contain some *recognition* information, since they can sometimes be used to eliminate candidate models.

Three Points: Further constraining $\mathbf{p}_3$ to lie on the model, for any sufficiently large object, the object is free to move as above, except that the rotation about the line connecting $\mathbf{p}_1$ and $\mathbf{p}_2$ is prevented. Thus three DOF remain. Note in Fig. 1 that the point $\mathbf{p}_1$ can still be slid anywhere on the model surface without violating the three point constraint. Thus we have not yet invoked much information about the object's shape. However, three points provide slightly more recognition information than two, since some objects fitting two given points might not be large enough to fit three.

Four Points: Further constraining $\mathbf{p}_4$ to lie on the model, we see in Fig. 1 that only two DOF remain. The four points are free to translate parallel to the x axis, and for a given x position of $\mathbf{p}_4$, they can rotate about the vertical line through $\mathbf{p}_4$. Note that with four points on the surface, we already have strong constraints on where the points may lie. For example, $\mathbf{p}_1$ may not lie in the shaded region and similar regions on other faces.

Thus, four points provide some shape information about the surface on which they lie, since they have this discrimination. This constitutes significant recognition information, since some objects cannot fit a given set of four points, even if the objects are large. Four points can be thought of as providing some surface curvature information; for example, a modeled sphere of only one particular radius fits four given (non-cocircular) points.

Five Points: p_5 prevents rotation about the vertical line through p_4, leaving only one DOF; translation parallel to the x axis. The recognition information is also stronger; e.g., most sets of five points don't fit any sphere.

Six points: Finally, constraining p_6 to lie on the surface as shown prevents all local relative motion between the model and the six points, since it prevents the x translation discussed above.

The above discussion shows that a small number (four to six) of range points can contain sufficient information to greatly reduce the set of possible identities and poses of the object from which they were sampled. In particular, in the absence of the degeneracy effects of object symmetries, the number of remaining DOF in object pose is 6–n for n range points, if the object is not eliminated as a recognition candidate. We will next consider some special classes of modeled objects whose symmetries allow easier elimination of candidate models during recognition.

1.3 Some Examples of Object Symmetries

Example 1; n points on a planar surface, $n > 2$. In this case there are three DOF, corresponding to rotation and translation in the plane, regardless of the value of n. This, along with the ubiquity of planar surfaces, makes recognizing planar regions of a object's surface the subject of specialized algorithms [8]. Note that in the case of zero uncertainty, four points are sufficient to either eliminate or provide strong evidence for a planar surface.

Example 2; n points on a spherical surface, $n > 2$. This is a generalization of example 1, with three DOF. Again, four points are sufficient to either eliminate or provide strong evidence for a sphere of given radius.

Example 3; n points on a cylinder, $n > 3$. There are two DOF, rotation about the axis and translation along the axis.

Example 4; n points at arbitrary places on a helical cylinder, $n > 4$. The "spring" shaped object can turn like a screw in one DOF regardless of n. The same numbers apply to the torus and the general prism (linear extrusion).

What These cases have in common is that the number of DOF is greater than 6-n. Specifically, as successive range points are introduced, the number of DOF is successively reduced until the number of DOF characteristic of the symmetry is reached, and it remains fixed as more range points are introduced. One implication of this is that it is possible to recognize certain objects with a high degree of confidence with a very small number of (sufficiently accurate) range measurements. For example, four points either decisively eliminate a sphere of a given radius, or provide strong evidence of a match. Also, only three of those points allow localization of the sphere, insofar as we don't care about the three rotational degrees of freedom associated with the symmetry. In the case of a cylinder of given radius, five points either decisively eliminate the object or provide strong evidence of a match. The knowledge that a certain four points lie on the cylinder suffices to determine its pose. Six points can eliminate a cylinder of arbitrary radius. To see this clearly, consider placing $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ on a cylinder and sliding the cylinder on them until $\mathbf{p}_4$ makes contact, if possible. This clearly can happen for a range of radius values, in general. But $\mathbf{p}_5$ will then lie on the surface only if the cylinder has one particular radius value. Once we hypothesize the radius that makes the five points fit, if $\mathbf{p}_6$ doesn't fit, all cylinders are eliminated. Many other cases are worthwhile to study, but are left to the interested reader.

1.4 Recognition with a Small Number of Range Points; Discrete Analysis

Now we will perform an exercise similar to that of section 1.2, representing pose combinatorially (and approximately) by the association of range points with model facets, instead of by six coordinates. We will use as the model a polyhedral approximation of the object, using a large (say > 1000) number M of facets, each of which is compact, so that its maximum linear dimension is small. The reason for this is to make the range of distances between a point on one given facet and a point on another as small as possible. This will then enable us to use the distance between two range points to maximal advantage in determining where on the surface of the model they could lie. We will use the term *surfel*, for surface element, to denote a model patch of this kind, with an upper bound placed on the ratio of the maximum linear dimension of the facet to the square root of its area, as a prescription for compactness.

Now consider choosing arbitrarily the set of range points $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6\}$ from the range image. As in section 1.2, we will temporarily ignore grouping and uncertainty considerations. We will make a crude approximate analysis of the combinatorics of interpreting four range points. Figure 2 represents a modeled object with six range

points indicated on its surface. We define $d_{ij} \equiv \| \mathbf{p}_i - \mathbf{p}_j \|$, the distance in space between two range points. In attempting to interpret point $\mathbf{p}_1$ we note that it could have come from any one of the M facets of the model. For each of these M 1-interpretations, say $(\mathbf{p}_1, f_1)$, $\mathbf{p}_2$ can lie only on one of the k surfels overlapping an approximately spherical shell surrounding surfel f_1. Specifically, this shell is the locus of points located a distance d_{12} from any point on the surfel f_1. Intuitively, k is typically considerably less than M. It appears that k is $O(\sqrt{M})$ in the absence of uncertainty. This is supported by the observation that as M increases, the facet density on the surface increases linearly in M, while the surface area subtended by the shell is proportional to $1/\sqrt{M}$, since the maximum linear dimension of the surfels, and hence the shell thickness, varies as $1/\sqrt{M}$. Thus there are usually considerably less than M^2 2-interpretations of two given points $\mathbf{p}_1$ and $\mathbf{p}_2$. The preceding arguments suggest $O(M^{3/2})$ 2-interpretations. Now for each 2-interpretation of $\mathbf{p}_1$ and $\mathbf{p}_2$, $\mathbf{p}_3$ can generally lie only on a member of a very small subset of facets, since $\mathbf{p}_3$ lies at known distances from both $\mathbf{p}_1$ and $\mathbf{p}_2$; d_{13} and d_{12}, respectively. Thus, it appears that we can typically expect $O(M^{3/2})$ 3-interpretations of three given points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$.

For many of the consistent 3-interpretation of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, the point $\mathbf{p}_4$ will typically not to be able to lie on any facet. This is because four is the smallest number of points that cannot be rotated and translated as a rigid body so that they can be made to lie on any (sufficiently large) surface. That is, they contain some information about the shape of the surface they lie on, as argued in section 1.2. When $\mathbf{p}_4$ *is* consistent with a particular 3-interpretaetation, the number of facets it may lie on will typically be very small. Thus, we can typically expect that the number of consistent 4-interpretations of four given points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ is less than the number of consistent 3-interpretations of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. We conjecture that in the absence of uncertainty and certain object symmetries, this fourth point reduces the number of consistent interpretations by a factor of $O(\sqrt{M})$, and that the same applies to $\mathbf{p}_5$ and $\mathbf{p}_6$, so that six points are consistent with only $O(1)$ interpretation. This is supported by the continuous analysis of section 1.2, in which each point reduces the number of DOF by one. Here, M is the number of values in a discretization of a 2-D surface. Therefore removing one degree of freedom in the continuous analysis should correspond to a factor of $O(1/\sqrt{M})$ in the discrete analysis. If we considered uncertainty here, then for sufficiently large M the shell would approach constant thickness, and the $O(1/\sqrt{M})$ factors would apparently be replaced by factors of c, where c is approximately constant and $c < 1$. See [1] for more thorough combinatorics.

The observations above suggest that if we could interpret at least four range points simultaneously, we could eliminate the need to generate all those 3-interpretations of

$\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ that were pruned away when $\mathbf{p}_4$ was processed, and generate only those few interpretations of $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$ that are consistent with the six pairwise distance constraints. We would like to make this process a simple table lookup, in which we use some simple numerical feature of the four points (such as the radius of the fitting sphere) as an index into a table whose entries are lists of quadruples of model facets matching the four respective points. However, there are M^4 such quadruples, a prohibitively large number. This can be reduced to approximately $O(M^{3/2})$ by restricting the relative positions of the four (or more) points as described in section 2.

2. Tripod Operators

2.1 Mapping Range Points to Model Poses

Having argued that a small number (four or more) of range points can contain a great deal of information about the identity and pose of the object on which they lie, we would like to exploit this with some efficient "points to poses" mapping procedure that can be applied to a range image. Here are some desired properties of this procedure, and some comments on how we obtain them:

1. It should be local, in that it operates only on range points within some sufficiently small region of 3-D space, to facilitate its application to data lying completely within a single object.
2. It should be computationally efficient. Therefore we want the mapping procedure to rely as heavily as possible on precomputed lookup tables.
3. It should require a practical amount of storage.
4. It should preserve as much information about pose and identity as possible from the original range points used, so that when it is used as part of some complete recognition/localization system, it will need to be performed a minimal number of times.

In succeeding sections we introduce a way of achieving these properties. Satisfying properties 3 and 4 simultaneously was a key issue. The difficulty is seen in Fig. 3a, in which the mapping is from points in $\mathbf{R}^{12}$ to regions of $\mathbf{R}^3 \otimes SO(3)$, which is intractable as a table lookup even though we are considering only four points, the minimum required to convey surface shape information. Figure 3b outlines a solution to this problem. The right hand side represents pose using the well known idea of interpretations, resulting typically in less than $O(M^{3/2})$ interpretations for M model surfels, for each set of four points. This parsimony stems from the separation of two kinds of pose information; that

of the set of points with respect to the rangefinder and that of the model with respect to the set of points, along with the knowledge that the points lie on the object.

The left side of Fig. 3b introduces a new idea; the structuring of range points during the process of selecting them by requiring them to satisfy various geometric constraints relative to one another so that for n points, there are n-3 free parameters describing the *relative* positions of the range points. These parameters are features corresponding to shape properties of the surface. This structuring is accomplished with the tripod operators described in the following sections. Thus in Fig. 3b a tripod operator first selects four appropriate samples in a given region of the range image and generates one real valued feature. This is then mapped via precompiled tables into a set of interpretations of the four points. Both steps are in constant time.

2.2 A Simple Four Point Tripod Operator

We will now define a specific tripod operator as an introductory example. Consider the following geometric object, illustrated in Fig. 4a; a set of three points, called *feet*, and one line arranged as follows: the points are at the vertices of an equilateral triangle of edgelength d, and the line, which we will call a probe line, passes through the center of the triangle and is perpendicular to its plane. We will denote by A, B and C the feet of any tripod operator. To apply this four-point operator, consider a two-dimensional surface imbedded in three-space. This will be in the form of a computer representation of a rigid physical object, such as a surface interpolation of a range image, or a surface model of an object obtained from a computer aided design system or from range images. Now imagine rotating and translating the operator as a rigid body until its three feet all lie on the surface, much as in placing a surveyor's transit or a camera tripod. Now if the probe line associated with the operator intersects the surface at a point denoted by D, the distance from D to the plane of the triangle can be regarded as a feature value generated by application of the operator to the surface. If A, B, C, and D fall at position vectors $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$, respectively, then

$$s = \frac{((\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1))\cdot((\mathbf{p}_3-\mathbf{p}_4)}{\|((\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1))\|}$$

can be used to compute the value of this tripod feature. Like all tripod operator features, s is an intrinsic property of the object represented by the surface; it depends on the shape of the object and on where the operator is placed on the object, but not on where the object and the points are located in any coordinate system. For example, a positive feature value represents a local depression in the surface, and a negative value represents a local "bump".

Now suppose that the object surface is modeled by M surfels as in section 1.4. Let us consider the ways that the operator can be placed on the various surfels of the model. Denote by A, B and C the three feet of the tripod operator, and the intersection of the probe line with the surface will be called D.

From the discussion of section 1.4, A, B, and C can typically have $O(M^{3/2})$ placements on respective surfels of the model, and for each, the probe line is nearly fixed, so that the intersection point D can fall on very few (O(1)) surfels. Thus the tripod operator can be placed on the model in $O(M^{3/2})$ ways, each yielding a feature value and an associated 4-interpretation, which we can store in a table indexed by the (discretized) feature value (see section 2.7). If the operator is later applied to the interpolated surface of a range image for the purpose of recognizing and locating that modeled object, it yields points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$, corresponding to the tripod operator feet A, B and C, and the probe point D, respectively, and a resulting feature value. This feature value then can be used to eliminate from consideration any interpretation not present in the table. If there is no table entry at all for a certain feature value, the whole model is eliminated, as in the case of a sphere of the wrong radius, for example. Thus the tripod operator can be a powerful feature detector for use in a recognition/localization system.

2.3 A General Class of Tripod Operators

We will now define a broad class of tripod operators. An n-point tripod operator consists of three points A, B, and C, with distances a, b, and c between them, as shown in Fig. 4b. Also there are $n-3$ space curves $\{\mathbf{x}_1(s_1), \mathbf{x}_2(s_2), ..., \mathbf{x}_n(s_{n-3})\}$ which we call probe curves. Each probe curve $\mathbf{x}_i(s_i)$ is a position vector as a function of a scalar parameter s_i, which represents arc length along the curve. The application of the operator to a surface results in the values of the $n-3$ scalars s_i determining where the probe curves intersect the surface. They can be regarded as forming a feature vector of length $n-3$ describing the local shape of the surface. An important general property of tripod operators is that for any modeled solid, applying an operator everywhere possible on its surface generates a manifold in the feature space with dimensionality not exceeding three. We can see this by noting that the three feet of a tripod operator can slide on a surface in three DOF, which parametrize the $n-3$ features. In cases of object symmetry the dimensionality of the feature space can be zero (sphere), one (cylinder), or 2 (torus, helix or extrusion). We define the *order* of an n-point tripod operator to be $n-3$; the number of scalar features generated.

2.4 Linkable Tripod Operators; the Four-Point Case

We now describe a class of tripod operators with particularly interesting properties. We will start with a four-point instance. The three feet A, B, and C of the tripod are at the vertices of an equilateral triangle of length d, and a probe curve is formed by a circle centered at the midpoint of the edge BC and coaxial with it, as shown in Fig. 4c. The radius is $\sqrt{3}d/2$, so that any point D on the circle is at a distance d from both B and C. When applied to a surface, four point operator returns one parameter value, the angle θ between the triangles ABC and BDC, where D is a point where the circle intersects the surface. Our convention is that $\theta = 180^\circ$ for a planar surface, with $\theta > 180^\circ$ if the hinge edge BC looks convex from the rangefinder's viewpoint. The application of the operator to a surface, yields $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ as the position vectors of A, B, C and D, respectively, and the scalar feature θ. Note that this operator has a bilateral symmetry. It is essentially two equilateral triangles joined by a hinge joint at their common edge, and after it is applied to a surface, it makes little difference which triangle is regarded as the tripod. This leads to the idea of making a second application of the operator at the three points $\mathbf{p}_2$, $\mathbf{p}_4$ and $\mathbf{p}_3$ on the surface of a range image, producing a new point $\mathbf{p}_5$ as shown in Fig. 5 and a new feature value θ'. Thus for the second application of the operator, A, B, C and D are at $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_3$ and $\mathbf{p}_5$, respectively.

Now note that we have succeeded in linking these operators together, so that we can combine the information gotten from their feature values. If we use the first operator application to look up the 4-interpretations of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ for some model, and the second to look up the 4-interpretations of $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_3$ and $\mathbf{p}_5$, we can retain the 5-interpretations consistent with both. This linking procedure can be repeated indefinitely. Figure 5 shows five operator applications, yielding eight points and five feature values. This example illustrates the opportunistic growing of links wherever they don't cross boundaries of image segments (see section 4, steps 1 and 7). One good mechanism for keeping track of these sets of consistent interpretations is interpretation trees (see Fig. 6), with the range points $\mathbf{p}_i$ as the sensor measurements and the surfels as the model elements, much as in [5]. The difference here is that the constraints among four measurements at a time are included at each new tree level, thus eliminating many branches without generating them. Also, the constraints are somewhat stronger taken among four points at once, since a 4-interpretation satisfying the six pairwise constraints separately might not satisfy them simultaneously, and the latter is enforced by the 4-point operator.

The linking could be done using one or two common points instead of three as described, but linking three points has the advantage of preserving rigidity; the distance between any two points in Fig. 5 is known to within the uncertainties arising from finite surfel size and measurement error. Next we will show that the contents of this section generalize in a reasonable way to $n > 4$.

2.5 Linkable Tripod Operators; More Than Four Points

We can generalize the 4-point linkable tripod operator to any number of points by attaching additional equilateral triangles to previous ones by hinge joints, as in the 6-point example of Fig. 4d. Thus points E and F are similar in function to D; after planting A, B and C on a surface, E and F are moved through their respective circular paths until they strike the surface, yielding three feature values θ_1 , θ_2 and θ_3 for this 6-point operator. This is a particularly interesting case, and so we will discuss its properties and present some experimental results on its application to synthetic range images in section 2.6. It is possible, however, to generalize it to many other kinds of n-point linkable operators by arbitrarily connecting equilateral triangles to various free edges with hinge joints. Note that such an operator is sequentially folded onto a surface as one wraps a gift. Thus it extends the class of operators described in section 2.3, since the circular probe curves are in general not fixed with respect to the initial triangle ABC defined by the tripod feet.

The preceding discussion suggests that an n-point linkable operator is similar to a set of 4-point linkable operators appropriately linked together in the manner of Fig. 5. The difference is simply that an n-point operator is to be applied as a whole without varying its structure, whereas a linked set of operators can be constructed in a flexible opportunistic manner on a range image. Interpretation data can be precompiled (see section 2.7) for the feature values of a single operator, while a set of linked operators requires explicit merging of the interpretations of its constituent smaller operators.

2.6 Experiments with the linkable six point operator

We have implemented a software system in C on a Sun SPARCstation which allows the generation of synthetic range images of various solid models and the application of the 6-point operator to them using the procedure of section 3.1. The solid models are in the form of unions and intersections of analytic solids represented in the form $f(\mathbf{x})<0$. The range data is generated by ray tracing, using binary search to find surfaces at zero-crossings of the functions $f(\mathbf{x})$, and golden mean search to determine whether various protrusions are hit or missed. The rays are projected through the points of a rectangular grid.

The range images are rendered on the CRT as in Figs. 7a,e,g. The simulated rangefinder was located 20 units above these models, from which the rendered mesh would appear rectangular. The rectangular projection grid was located midway between

the rangefinder and the model center. The renderings use one curve per five range pixels. The 6-point operator can be placed either under manual control or automatically with uniform random distribution in the image index coordinates. There are three free variables here, as described in section 3.1. The other six figures show resulting points in the feature space spanned by angD $\equiv \theta_1$, angE $\equiv \theta_2$, angF $\equiv \theta_3$. The vertical bar indicates an interval of angF for which points are displayed in a projection onto the plane of angD and angE. Note that this operator posseses a 3-fold symmetry about the axis angD=angE=angF, although it would be easier to visualize if our projections were along that axis in the displays.

Figure 7a shows a 90° concave dihedral with one operator placement displayed. Figure 7b shows the feature points from 10,000 placements. Note that the display is really that of a 5-point operator (two features), since angF is projected. However, in Fig. 7c a slice of the point cloud at angF = 136° shows the 2-D nature of the manifold in feature space generated by this extruded shape, consistent with the discussions of sections 1.3 and 2.3. Figure 7d shows points with angF > 180°. The example placement in Fig. 7a produces such a point. note that the upper operator probe (F) is in the groove so that angF > 180°. this forces the lower left and right probes to climb the planes, so that angD and angE are less than 180°, as seen in the data of Fig. 7d. Figure 7e shows a half cylinder of radius 1 on a plane. Figure 7f superimposes data from random placements of operators of three edgelengths; d = .4, .6, .8 (We could have varied the cylinder radius for the same effect). Each d value yields a distinct 1-D oval curve for placements falling on the cylinder. The placements falling partly on the plane give 2-D data, and those entirely on the plane give the center point (180°,180°,180°).

The cylinder data is consistent with sections 1.3 and 2.3; from section 1.3, we can slide the cylinder in two DOF with respect to the six points without breaking contact or changing the θ_i values. Therefore, manipulating the remaining degree of freedom, corresponding approximately to rotating the operator within the plane of its feet, will generate a one-dimensional region in feature space. Note the small fraction of feature space occupied by these two symmetrical examples (actually zero for perfect data, small for realistic measurement errors from state-of the-art rangefinders). The cylinder example confirms the argument in section 1.3 that five points can eliminate or provide strong evidence of a cylinder of given radius, since the oval is sparse in two dimensions. Also, in the cylinder case, since the projection of the data onto angD spans a limited angle, four points can sometimes eliminate a cylinder, e.g., when angD alone is outside that range. This latter point applies to some extent to all objects.

Figure 7g is a half ellipsoid lying on the plane. Its three semi-axes are $\sqrt{.25} = .5$, $\sqrt{.4} = .632$, and $\sqrt{.6} = .774$. The operator edge length d is .4. The dark lower right cluster in Fig. 7h is from placements completely on the ellipsoid and is three-dimensional, as expected from the lack of appropriate object symmetries. Finally, Fig. 7i shows a 2-D manifold generated from a sinusoidal "washboard" shape, like a corrugated roof. The period is $\pi/5 = .628$, amplitude = .1, and d = .25 for the operator. Again the extrusion yields two DOF in feature space.

2.7 A Data Structures Relating Tripod Features to Models

The use of tripod operators in a vision system requires the efficient access of model information pertinent to given tripod feature values. This can be achieved by constructing in offline computations the data structure illustrated in Fig. 8. It consists of an array indexed by the feature values of the tripod operator. For example, in the case of the six-point linkable tripod operator, the three angular features are discretized at a resolution appropriate for the sensor uncertainty of our rangefinder and the resolution of the surface model. They can then be used as indices into a three index array. Each array element consists of the number of models which can possibly produce the given feature values, given noise tolerances, the average number of consistent placements over these models, and a pointer to the set of these models. This set is is an array whose elements give the name of a model (an integer), which points to the number of tripod placements on that model that could have produced the feature values to an array containing the set of placements of the tripod operator on the model consistent with the feature values.

3. Computation of Tripod Operator Features

3.1 Calculation of Tripod Operators on Range Images

Since the efficient application of tripod operators to range images is crucial to their effective use in a vision system, we will present a fast algorithm for doing this. We will treat the case of linkable tripod operators. We assume that a range image is given, along with formulas relating the coordinates of an arbitrary point in space with the two pixel indices of the range image. In a nutshell, the procedure finds the intersection between a test curve and the range image by binary search along the test curve until the distance between the some point on the test curve and the corresponding range surface point is sufficiently small.

We denote the range pixel whose horizontal and vertical indices are i and j, respectively, by the 3-vector $\mathbf{r}_{ij}$. This vector is given in a coordinate system in which the

viewpoint of the rangefinder is at the origin. We define $\mathbf{r}(h,v)$ as an interpolated range image such that $\mathbf{r}(h,v) = \mathbf{r}_{ij}$ if h=i and v=j. For non-integer values of h and v we will use triangulated polyhedral interpolation. Each i,j pair will yield two triangular facets; one with vertices at the range pixels (i,j), (i+1,j), and (i,j+1), and one with vertices at (i+1,j), (i,j+1), and (i+1,j+1). We denote by h(**x**) and v(**x**) the real valued functions mapping an arbitrary point **x** to the respective parameters of the corresponding point on the interpolated range image. That is, the ray from the origin of the rangefinder through **x** also passes through the range point $\mathbf{r}(h,v)$, where h = h(**x**) and v = v(**x**).

Now, to place a tripod operator on the interpolated range image, we first place point **a** of the operator at an arbitrary range point. Then we chose an arbitrary direction in the hv plane and search for a point **b** lying on the interpolated range image at a euclidean distance d from point **a**. We do this by binary search along a circle of radius d centered at **a** for a point for a point **b** whose z component equals that of $\mathbf{r}(h(\mathbf{b}),v(\mathbf{b}))$. The circle is oriented so that it is viewed edge-on from the rangefinder origin.

The third point **c** must be at a distance d from both **a** and **b**. It is calculated by binary search along a circle of radius $d\sqrt{3}/2$ centered at (**a**+**b**)/2 for a point for a point **c** whose z component equals that of $\mathbf{r}(h(\mathbf{c}),v(\mathbf{c}))$. The circle is oriented coaxially with the line through **a** and **b**. Any further points in a linkable tripod operator can be computed in exactly the same way; by chosing two existing points and searching along the circle that symmetrically bisects the line segment joining them.

Note that although there are plenty of pixels to chose from in a typical range image, the tripod operator choses only points related as described above, so that interpolated points between range pixels are often selected. We will see that this slightly awkward procedure is very well compensated for by the operator's advantages.

3.2 Calculation of Data Structures from Models

Many of the uses of tripod operators that we discuss require the availability of a data structure (see section 2.7) which relates tripod feature values to the possible placements of the operator on various models. This data structure is to be computed offline, so that the processing time is not so critical, but still care must be exercised to avoid prohibitive computation time. We will outline here one possible way to process a given triangle-faceted polyhedral model with the four point linkable operator described in section 2.4.

1. Find the set S of all pairs of facets such that there exists a pair of points, one in each facet, separated by a distance d. This can be done in $M(M-1)/2$ steps. Store these so that each facet is a pointer to an ordered list of the $O(\sqrt{M})$ facets located a distance d

away. Altogether, there will be typically $O(M^{3/2})$ pairs in S, from the discussion of section 1.4.

2. Find the set Q of all quadruples of facets such that five of the six pairs of facets in the quadruple are in S. The remaining pair will correspond to A and D from figure 4c. To do this, associate with A and B, respectively, each pair of facets in S. For each association, find facet C by intersecting the set of facets at a distance d from facet A with those at a distance d from facet B. Since these two sets are ordered, their intersection can be found in linear time, yielding $O(\sqrt{M})$ time to find facet C for each A,B pair. Then D is computed analogously to C. Thus Q can be computed in $O(M^{2})$ time.

3. For each quadruple in Q, we must place the four points of the operator on the four corresponding facets in a representative number of ways to obtain an estimate of the range of values of θ obtainable. An exact computational geometry approach seems impractically complicated, so we will use a sampling approach. Denote by $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ the vertices of a facet. Then points sampled from that facet can be generated by

$$\mathbf{p} = \mathbf{v}_1 + u(\mathbf{v}_2 - \mathbf{v}_1) + v(\mathbf{v}_3 - \mathbf{v}_1),$$

where

$$0 \le u \le 1,\ 0 \le v \le 1, \text{ and } u+v \le 1.$$

If the scalar parameters are varied in steps of .25, for example, we get 15 sample points. Then the four operator points can be placed on those sample points in the four respective facets which are mutually separated by $d \pm \varepsilon$ (except of course, A and D). The tolerance ε is intended to accommodate uncertainty due to imperfect modeling, imperfect range measurement and the spacing between facet sample points. It should be large enough so that every possible association of a θ value and a 4-interpretation is covered, to ensure no missed hypotheses when performing recognition and localization.

4. Use of Linkable Tripod Operators in a Vision System

We will now outline a particular way to build a recognition and localization system using tripod operators in conjunction with other techniques. This exercise will illustrate what properties of the operators are expected to have the most practical value. There are many design choices in such a system, and so don't claim that the choices are optimal in this example.

The vision problem addressed is as follows. We start with a set of N rigid objects for which we have triangle-faceted polyhedral models. We now are allowed to perform some offline processing of the models, producing data structures that will facilitate processing at recognition time. The system is then to be presented with a dense range image of a scene containing some subset of the M objects in arbitrary configuration. Then it is to recognize and localize as many of the objects as it can, as rapidly as possible.

First, in an offline process, the models are processed with six-point tripod operators to yield data structures relating tripod feature values with possible placements of tripod operators on the various modeled objects, as described in section 3.2. Then, when a range image of a scene is to be interpreted, execute the following steps:

1. Subject the range image to a grouping or segmentation procedure which results in the labeling of each pixel as a member of a region. We want these regions to have the property that no two pixels in the same region lie on different objects. Also, we would like to have as few as possible regions lying on any given object. Some good cues to boundaries between regions are depth discontinuities and concave slope discontinuities. Methods for range image segmentation are treated elsewhere [7,8,9].

2. Place the tripod operator at a number of random locations in the range image, retaining only those placements for which all six points lie in the same region (from step 1). For each placement, look up the number m_c of models consistent with its feature value. The phrase *look up* refers here to the data structure computed offline.

3. Select for further processing the region R containing the placement with minimal m_c, as long as $m_c \neq 0$.

4. Place the tripod operator at a number of additional random locations within the region R. Look up the set of models consistent with each placement and compute their intersection I. If I is empty, mark the region R "inconsistent with models" so that it will not be processed further, and go to step 3.

5. Select a model *M* from the set I, and look up the number of consistent interpretations on *M* for each tripod placement we have made in R. Select the placement P1 having the smallest number of consistent interpretations. This can be thought of as the automatic selection of a distinctive local feature, as opposed to using predefined specific kinds of features [10].

6. Look up the set S1 of interpretations of P1 on *M*, and express them as an interpretation tree. The depth of this tree is six, since we are using a six-point tripod operator.

7. Find a new placement P2 of the tripod operator in R such that P1 is linked to P2 via three common points, and look up the set S2 of consistent interpretations of P2 on M. Delete those paths in the interpretation tree giving interpretations inconsistent with the interpretations S2. Extend the interpretation tree to represent the constraints of S2. The interpretation tree now has depth nine.

8. Repeatedly link new operator placements to the existing ones until one of the following conditions is true:

a. The interpretation tree is empty (this model is inconsistent with these points); go to step 5 with a different model *M* from I.

b. A computation budget is exceeded; go to step 3 for a different region R.

c. The number of partial interpretations on *M* is less than some prescribed constant; do model test for each interpretation, using lots of pixels. Do gradient descent on best interpretations. If no survivors, go to step 5 to get a new M from I; else label R with the candidate interpretations and go to step 5.

9. *Go to step* 3 *if uninterpreted* regions remain; else terminate.

Note that a model test for k pixels can be done in O(k) time independently of object complexity if a fast proximity model of the object is available, such as a voxel model storing distances from every point to the model, at some cost in storage. More compact proximity models might be feasible using a few stored analytic distance formulas indexed by location. Also note that in the case of an operator placing points on more than one object, intuition suggests that this can frequently be caught by a failed interpretation, as supported by [1], or by the model test.

5. Conclusions and Future Directions

We have introduced a new class of operators for range images and made various arguments about their properties. We have described computational procedures for applying the operators to models and range images and outlined ways to use them in a vision system. They were experimentally applied to simulated range images, verifying some of the properties predicted.

Tripod cperators were shown to generate manifolds of dimensionality not exeeding three in feature space of any order. They provide a way to measure in constant time the distinctiveness of a local region of a range image, in terms of both the number of models

eliminated and the number of placements on the models eliminated. The former is expected to lead to approximately O(log(M)) time screening of a library of M objects, until a set of locally similar objects is reached. Tripod operators make the point-to poses mapping problem amenable at least partly to treatment by lookup tables, and are compatible with the method constrained search of interpretation trees, allowing the use of other constraints along with the tripods.

These operators suggest a great variety of future work. Their use in a complete vision system needs to be studied experimentally. Many specific properties, such as the amount of overlap in feature space between various objects, and dependence on sensor noise, need to be tested. These and the distinctiveness of feature values might well be studied information-theoretically. Traditional statistical pattern recognition might be very effective for model-free classification, since tripod operators generate low-dimensional, highly informative feature vectors. For example, a torus-like discriminate surface in a 3-D feature space could detect a cylinder. For higher order feature spaces than three, lookup tables are not even feasible, and analytic approximations of the 3-D subspaces for various objects might be very effective. This could lead to extremely fast recognition; eight points can be very discriminating for high precision range data, and their resulting five feature values might be tractable, since only 3-D subspaces need to be characterized. Also, mechanical tactile tripod operators might enable very fast tactile recognition.

Some flexible objects might be recognizable with some variant of the tripod operator, since when linked via three points they enforce local shape constraints more strongly than global ones, thus providing a potential method of approximating the continuum mechanics of bending an object.

In the near future, we plan to generate for various tripod operators, modeled objects, and amounts of noise the set of possible interpretations consistent with each value of the feature vector for that operator. This will then allow us to better answer such questions as how accurate a rangefinder is required for various recognition problems, what kind of tripod operators are most useful, how fine a surface tessellation is required in the model, and what speedup over the pure interpretation tree approach is provided. We also will study the scale problem; how many sizes of operators need to be used for a given library of objects, and how they are best used together in a system. We will study these problems in the context of building a high performance recognition system.

References

[1] Grimson, W. E. L., The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search, MIT AI Memo No. 1019, February, 1988.

[2] Gaston,P. C., and Lozano-Perez, T., Tactile Recognition and Localization Using Object Models: The Case of Polyhedra On A Plane, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6 (3):257-265, May 1984.

[3] Oshima, M. and Shirai, Y., Object Recognition Using Three-Dimensional Information, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(4):353-361, July, 1983.

[4] Faugeras, O.D., and Hebert, M., A 3-D recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces, Proc. Eighth Int. Joint Conf. Artificial Intelligence, pp 996-1002, August, 1983.

[5] Grimson, W.E.L., and Lozano-Perez, T., Model-Based Recognition and Localization from Sparse Range or Tactile Data, The International Journal of Robotics Research, Vol. 3, No. 3, pp 3-35, Fall 1984.

[6] Rioux, M., Blais, F., Beraldin, J., and Boulanger, P., Range Imaging Sensors Development at NRC Laboratories, Proc. of the Workshop on Interpretation of 3-D Scenes, pp 154-160, Nov. 27, 1989, Austin, TX, IEEE Press.

[7] Yokoya, N., and Levine, M.D., Range Image Segmentation Geometry Based on Differential Geometry: a Hybrid Approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11 (6):643-649, July, 1983.

[8] Faugeras, O.D., Hebert, M, and Pauchon, E., Segmentation of Range Data into Planar and Quadratic Patches, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp 8-13, June, 1983.

[9] Han, J.H., and Volz, R.A., Region Grouping from a Range Image, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp241-248, June 1988, IEEE Press.

[10] Bolles, R.C., and Cain, R.A., Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method, International Journal of Robotics Research 1(3):57-82.

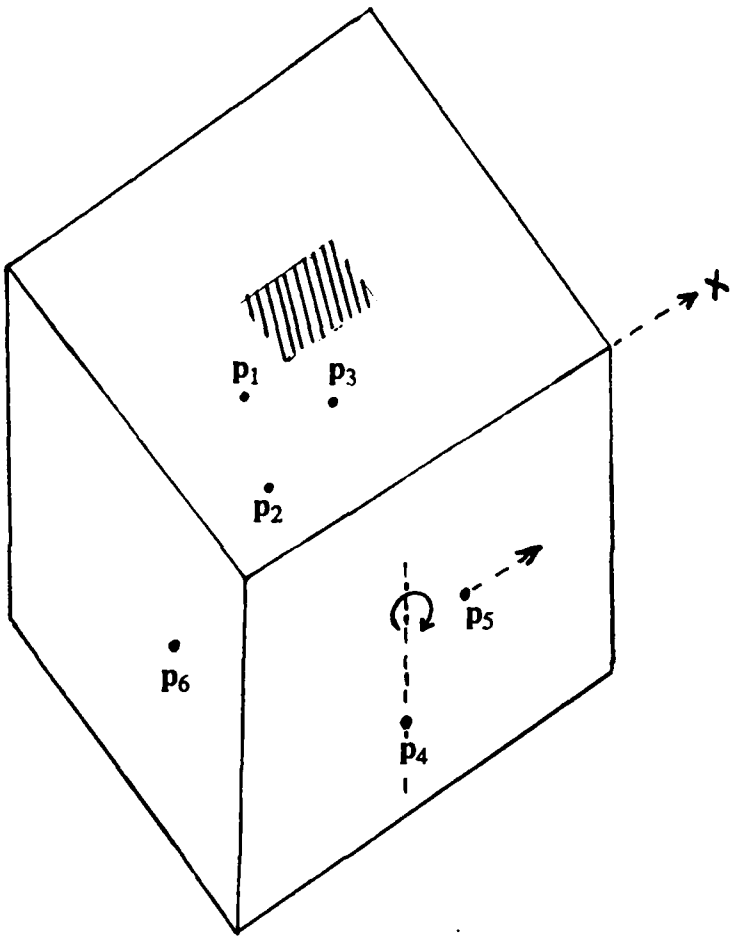


Fig. 1 — Six Points Placed on a Cube. Requiring each to lie on the cube successively reduces the number of DOF from 6 to 0.

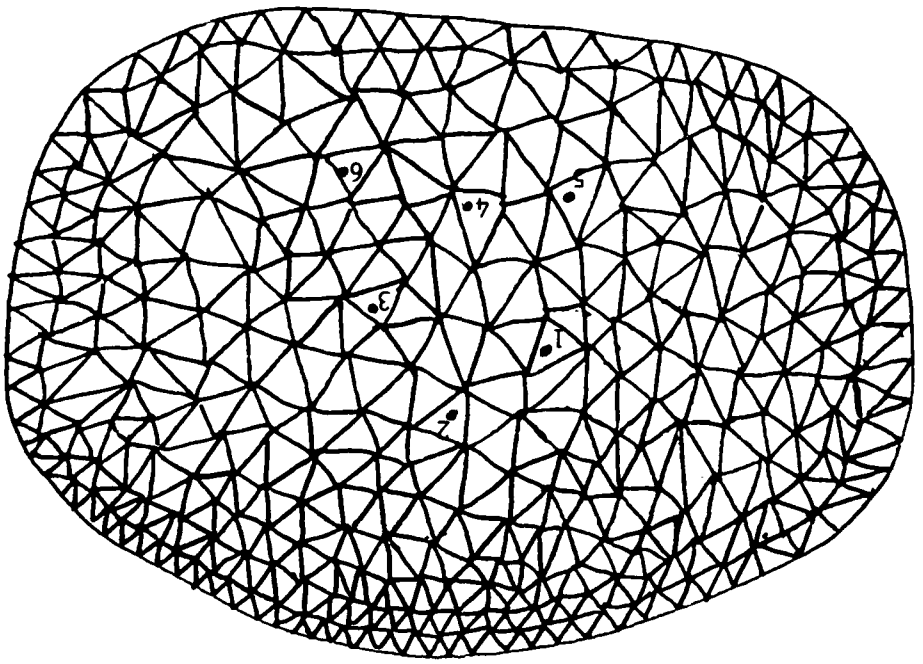


Fig. 2 — Six range points on a finely and compactly triangulated surface model, with M surface elements (*surfels*). We Conjecture that in the absence of uncertainty and certain symmetries, three points have $O(M^{3/2})$ consistent interpretations, four points have $O(M)$, five points have $O(\sqrt{M})$, and six points have $O(1)$.

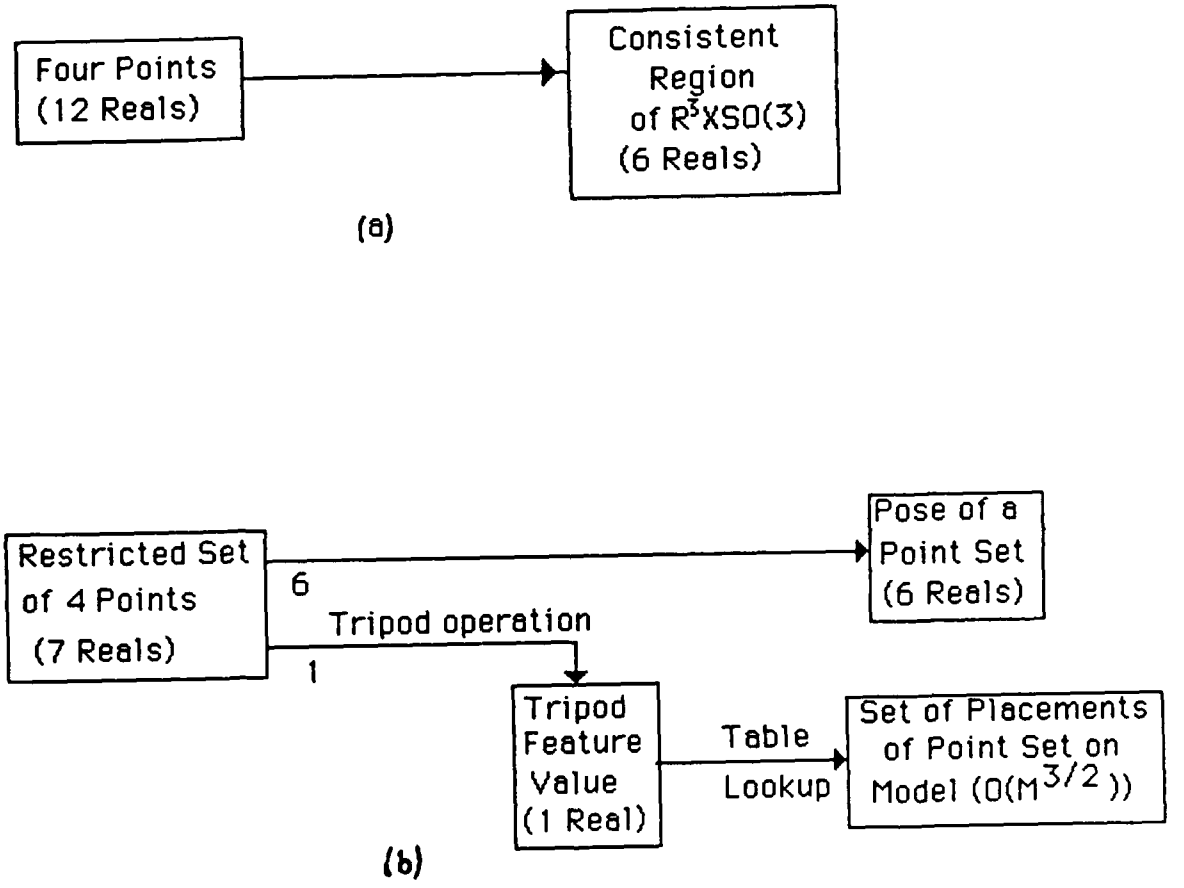


Fig. 3 — Dimensionality considerations in mapping points to poses; four point case. (a) Table lookup approaches intractable. (b) Tripod operators reduce dimensionality by structuring points and separating pose from local shape information

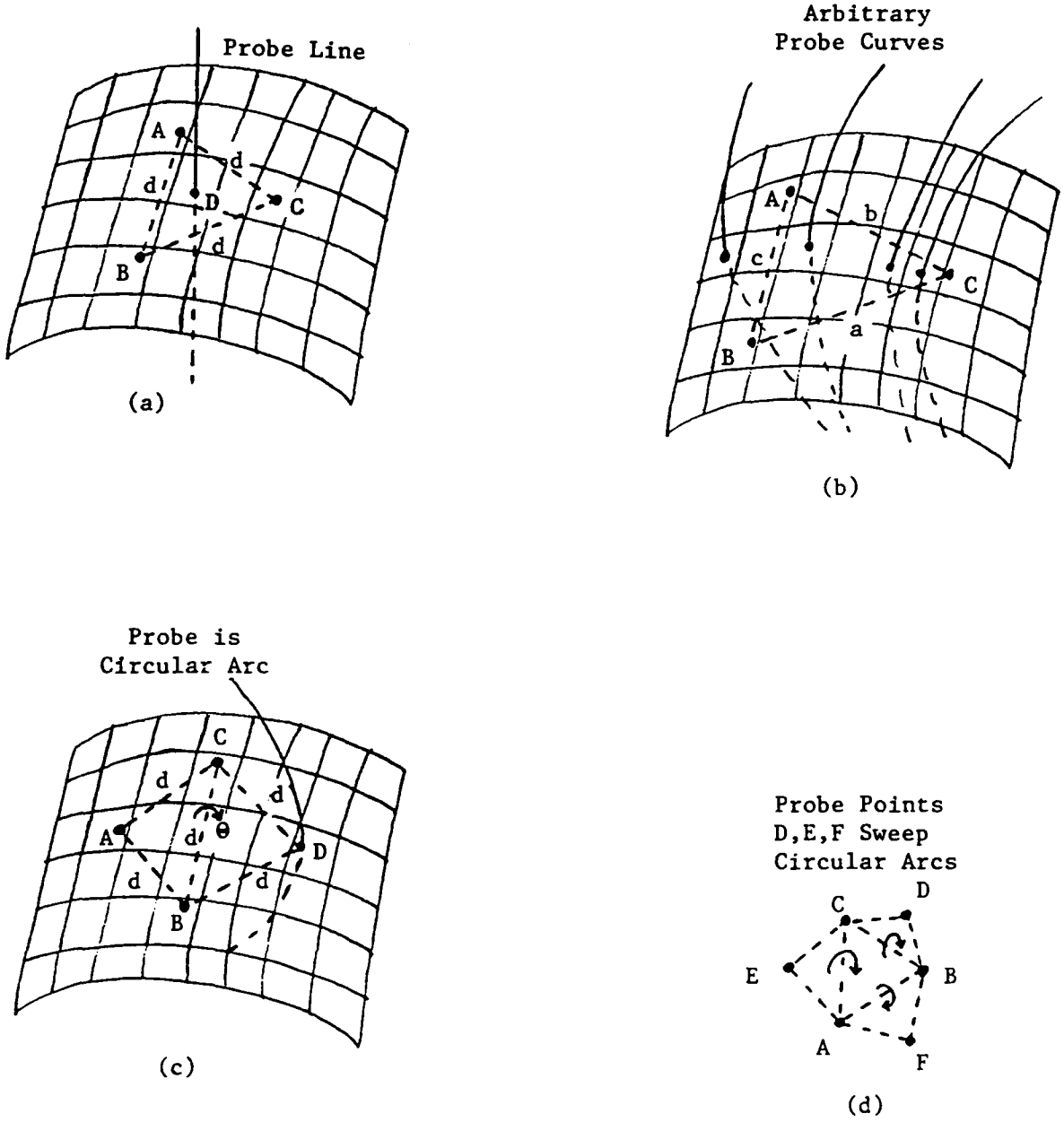


Fig. 4 — Examples of tripod operators; (a) Simple 4-point. (b) General. (c) Linkable 4-point (1 feature). (d) Linkable 6-point (3 features).

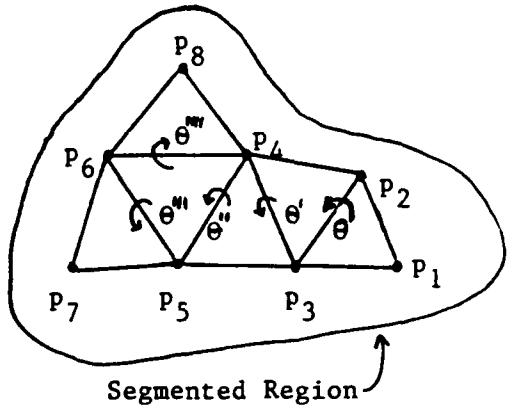


Fig. 5 — Linkable five 4-point operator placements in order to efficiently find the 8-interpretations consistent with both

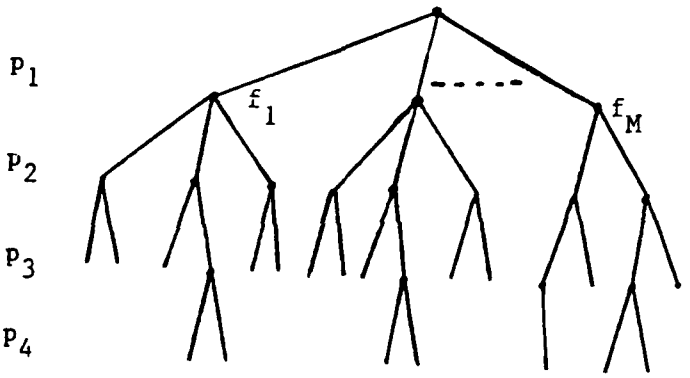


Fig. 6 — Review of the use of the interpretation tree to find matches of range points $\mathbf{p}_i$ to model patches (surfels) f_j such that geometrical relations among the $\mathbf{p}_i$ are consistent with those among the corresponding f_j. A tree node is a pairing $(\mathbf{p}_i, f_j)$.

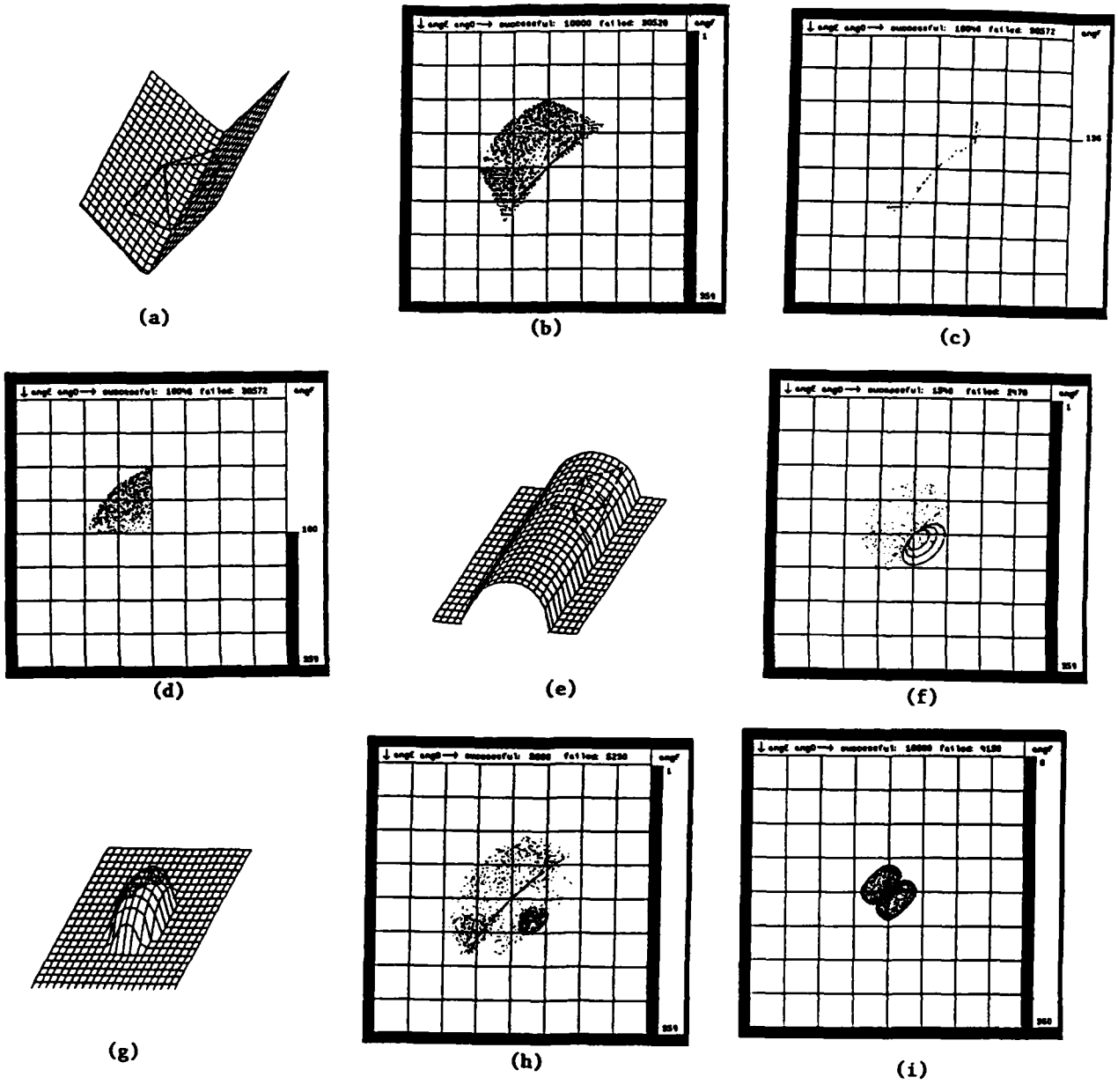


Fig. 7 — Data: 6-point operators applied to simulated range images. (a) 90° dihedral, with operator. (b) Full feature space for (a). (c) Slice at angF = 136° showing that (a) yields 2-D region. (d) angF > 180° implies angle D & angE < 180° in (a). (e) Half cylinder on plane. (f) Cylinder yields distinctive 1-D region (oval space curve) for each of 3 d values. (g) Ellipsoid (h) Dark lower cluster is 3-D region for ellipsoid. (i) Feature data for a sinusoidal "washboard" surface.

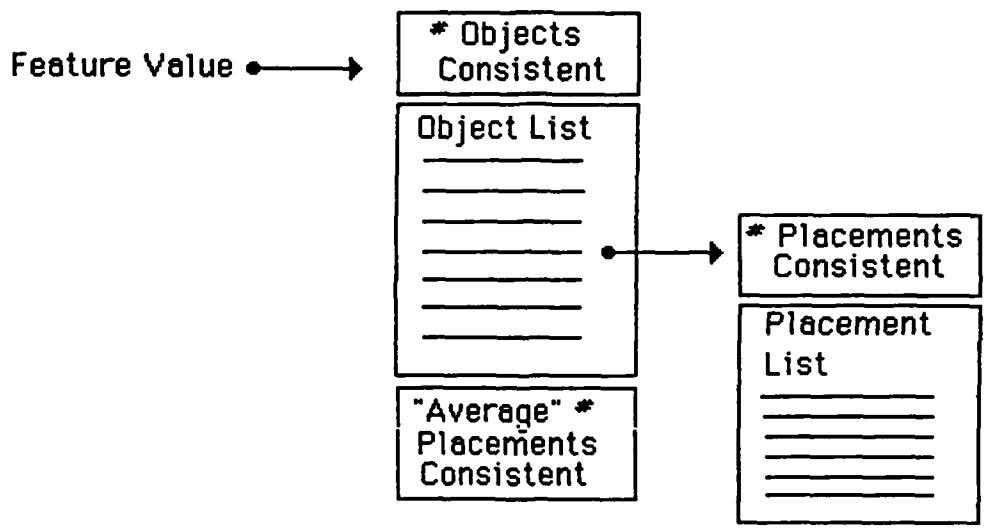


Fig. 8 — Data structures indexing pose and identity information by tripod feature values, to be computed offline from models, and used during recognition/localization.